EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

A Secure and Efficient Blockchain-Based Framework for Smart Cities Using Physics-Informed Neural Networks

Mohd. Asif Gandhi[1,*], Atul D Narkhede[2], N. Noor Alleema[3], M.P. Indumathi[4], Deepak Sundrani[5], R. Sasikala[6]

[1]Department of Mechanical Engineering, School of Engineering and Technology, Anjuman-I-Islam's Kalsekar Technical Campus, Panvel
[2]Department of Computer Engineering, Universal AI University Karjat, Maharashtra, India
[3]Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India
[4]Department of Science and Humanities (Chemistry), R.M.K. College of Engineering and Technology, Puduvoyal, India
[5]School of Construction, Nicmar University, Pune, India
[6]Assistant Professor, Department of MCA, Erode Sengunthar Engineering College Perundurai, India

Abstract

The massive scale and extensive implementation of the Internet of Things (IoT) makes it difficult to provide secure and private communications over it. Privacy and decentralisation have been made easier using blockchain technology. Unfortunately, these solutions aren't practical for the majority of IoT uses because of how much time and computing power they require. Secure and private IoT that makes efficient use of available resources is proposed in this study. With the use of Physics Informed Neural Networks, the technique takes advantage of the computing power available in IoT settings like smart cities. This solution examines the reliability of the blockchain-based Smart Cities Architecture with respect to accessibility, privacy, and integrity. When weighed against the security and privacy advantages our system offers, our simulation findings reveal that the overheads (distribution, processing time, and energy usage) are negligible.

Keywords: Physics Informed Neural Networks, Smart Cities (SC), Block Chain, Neural Networks, Kernel Principal Component Analysis (KPCA), Feature Selection, Normalization, Privacy-Preserving and Secure Framework (PPSF)

Received on 05 November 2024, accepted on 30 March 2025, published on 14 April 2025

doi: 10.4108/eetiot.7740

Introduction

The term Smart city can refer to a variety of urban planning strategies. All of them, nevertheless, have one thing in common: they have to change to fit the people who use them. Customers of smart cities need services that are Invisible and have practical uses. A wide range of H2H, M2M, and H2M interactions and applications are made possible by the technology of smart cities. By linking sensors, gadgets, and computers, the IoT makes Big Data collection and processing possible, paving the way for Smart Cities. When things can be recognised, digitally modelled, and connected by sensors and wired or wireless networks, we say that they are part of the IoT. A logical progression from decentralisation, transmission networks, communication protocols, cloud computing, and edge computing is the IoT.

*Corresponding author. Email: masifigandhi@gmail.com

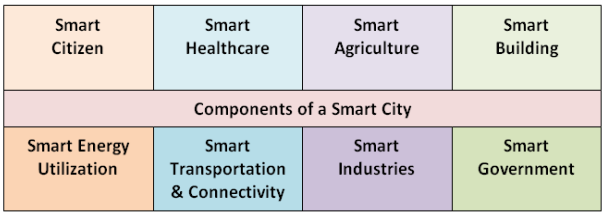

Figure 1. Key elements of an intelligent urban environment

The fundamental components of a smart city are illustrated in Figure 1. Several benefits, including asset optimisation, energy efficiency, and maintenance, are made possible by the IoT in smart cities, which allow for intelligent decision-making and improved services [1]. Cyber-attacks against smart cities include Ransomware, DoS, Man-In-The-Middle, and DDoS [2][3]. Data security, reliability, and accessibility are at risk from ever growing dangers. Therefore, SCs and linked networks must prioritise user privacy. It is believed that the architecture that combines privacy, security, and trust is adequate to handle the present challenges. Establishing trust and providing correct outcomes requires identifying and fixing unsecure and faulty IoT nodes [4]. Addressing security issues may result in high computation costs for ML-based IDS models [5], privacy violations in Cloud-IoT applications [6], and false alarm rates for IDS models [7]. To ensure trustworthiness among IoT nodes, smart cities must be developed holistically. In response to these issues, this research proposes a Physics Informed Neural Network that is blockchain enabled in order to safeguard SCs' privacy, security, and secrecy [8]. Privacy-preserving methods, including differential privacy and homomorphic encryption, were implemented to safeguard data sharing among stakeholders. The immutable ledger of blockchain enhances accountability and trust. For smart cities that are enabled by the IoT, blockchain technology can create a distributed cloud, which can support cloud computing. With this method, users of the stage who are connected to the IoT may verify the functionality in real time, and the cloud infrastructure can be identified and managed. It can also be held responsible for its actions. Here is how the rest of the paper is structured: Research on smart city applications of deep learning and Blockchain was covered in Section 2. This investigation's methodology is detailed in Section 3. Results and analysis of the investigation's performance are presented in Section 4. Finally, Section 5 suggests future studies on the topic to round out the study.

Literature Survey

Given blockchain's practical benefits, the literature stresses the need to incorporate technology with business logic. Without a trusted third party or administrator, a decentralised network known as a blockchain can function. To make smart city apps more secure, researchers have built frameworks that use blockchain technology. Due to its immutability, distributed ledger technology is favoured for the implementation of trustworthy algorithms in smart city data exchange. Applications in smart environments prioritise privacy. One important step towards making smart cities more private is PrivySharing[9]. In order to study and derive insights from IoT data in a way that safeguards privacy, the authors suggest a blockchain-enabled architecture that makes use of machine learning. To keep sensitive information private while it's being processed and analysed, it employs techniques like differential privacy, secure multi-party computation, and homomorphic encryption. Nevertheless, in order to protect individuals' privacy, differential privacy introduces noise into data [10]. Derivative insights may be less accurate and meaningful patterns may be more difficult to uncover if there is noise. Following previous work, the proposed method employs non-interactive zero-knowledge (NIZK) proofs to permit anonymous authentication of IoT devices while preserving privacy. This function prevents tracking and profiling by enabling devices to authenticate with the operation centre without disclosing their identities. The key agreements between the operation centre and verified IoT devices are further secured using bilinear pairing-based cryptography. Making a revocation system that can delete malicious entities while protecting the anonymity of legitimate users shouldn't be the primary goal [11]. A secure SVM-based solution using blockchain was developed by the authors [12] to address the issue of gathering training data from various providers. Secure building blocks are created by encrypting sensitive data from IoT devices using a homomorphic cryptosystem. Another study found that many different types of applications were more secure when blockchain and AI were combined. Federated learning, machine learning, and blockchain all work together to improve data sharing among IoT devices. FL systems are great at protecting sensitive information and avoiding data loss. The data is exchanged with the learning model by the system. The data utility of the model has to be enhanced [13]. The authors unveiled a system for managing access to and exchanging data that is based on blockchain technology. For the purposes of user identification, network administration, and behaviour detection, this article suggests a plethora of smart contracts. In order to identify user misconduct, the punishment was implemented. The experiment greatly reduced execution costs and increased data exchange security, according to the researchers [14]. The study [15] provides a smart healthcare architecture that uses blockchain technology and an ODLSB model, which is based on optimal deep learning, to facilitate the IoT. A secure blockchain approach based on deep learning is presented in this study. [16] propose a crowdsourcing FL system that uses blockchain technology to better understand customers. In order to secure the IIOT within a TEE, Aditya Pribadi and colleagues have created a system that employs FL technology based on the blockchain. By taking the effect of model correctness into account, their method improved federated learning's privacy and security. [17] 5G network trials were suggested. The author used

smart contracts as an alternative to a central server to authenticate UAVs that fly across domains and aggregate models. [18] established a method for the safe classification of images and communication amongst UAV networks. Clustering, private communication with blockchain, FL picture categorisation, and performance validation using pre-trained CNN models are the three parts that make up the framework. While many research have looked at blockchain technology as a potential solution for UAV-based FL, very few have taken into account the following: aggregator selection based on reputation, global model aggregation, validation using pre-trained CNN models, and benchmark datasets [19]. A horizontal FL powered by blockchain technology for unmanned aerial vehicles. The proposed architecture integrates strong encryption for data transmission and a blockchain-based ledger for transaction integrity, reducing the dangers of data poisoning and inference attacks. Anomaly detection algorithms were incorporated to proactively identify and address suspicious activity in IoT environments.

In order to get high-quality TSEs for smart city networks using minimum observational speed data, this research provides a PINN framework. To overcome their shortcomings and capitalise on their strengths, PINNs combine model-driven and data-driven approaches. Results from experiments show that the suggested approach resolves smart city network problems with high accuracy.

Proposed Method

The proliferation of the IoT has contributed to the rise of the smart city concept. IoT networks allow Internet-connected smart city equipment to gather and process data. Centralisation, security, privacy (e.g., inference attacks and data poisoning), transparency, scalability, and verifiability are some of the challenges that slow down the development of smart cities [20]. In light of the above, we offer a Privacy-Preserving and Secure Framework (PPSF) for smart cities powered by the IoT. The use of PINNs improves precision by integrating physical rules into the learning framework, thus diminishing dependence on extensive labelled datasets and enhancing generalisation. The PINN framework attains computing efficiency by directly solving PDEs, hence minimising computational overhead relative to conventional iterative methods.

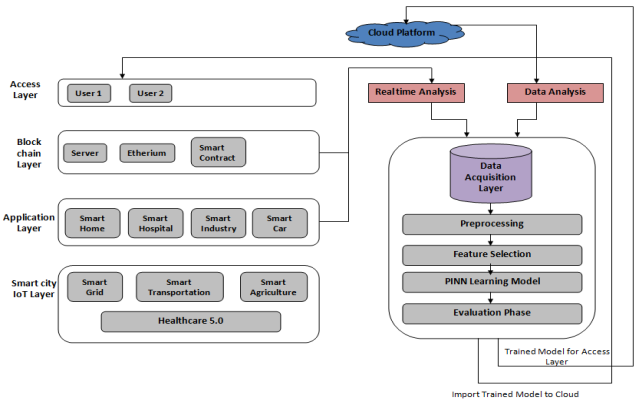

Figure 2. A four-tiered smart home application architecture that utilises the PINN Learning Model and is built on the blockchain

For blockchain-based smart homes, Figure 2 depicts the proposed architecture of Physics-Informed Neural Networks (PINN). The resilience of the blockchain architecture was confirmed through stress-testing scenarios including dynamic traffic patterns. Methods like dynamic node allocation and redundant data paths guaranteed system dependability.

Feature Normalization

IoT sensors generate data of varying sizes. By implementing a min-max normalisation strategy, the PINN framework may reduce bias in IoT network traffic while maintaining statistical data. The technique involves setting the lowest value to 0 and the greatest value to 1, and then converting all other values to decimal points between those two extremes. The transformation function can be applied using Equation (1).

$$s_{new} = \frac{s - s_{min}}{s_{max} - s_{min}} \quad (1)$$

Identifying the highest (s_{max}) and lowest (s_{min}) values of a characteristic in IoT network traffic is necessary for scaling it down.

Feature Selection

To minimise performance deterioration in a smart city environment, feature selection is used to identify relevant features and remove unnecessary ones to generate a subset that accurately depicts the situation. One elementary statistical tool used by the proposed TP2SF architecture is the Pearson correlation coefficient (PCC). This method finds the degree to which two variables are comparable [21]. The TP2SF architecture changes the most important features, those with the lowest N rankings. Calculate PCC using Eq. (2) for two features s_1 and s_2.

$$PCC(s_1, s_2) = \frac{\sum_{p=1}^{x}(n_p - \bar{s_1})(k_p - \bar{s_2})}{\sqrt{\sum_{p=1}^{x}(n_p - \bar{s_1})^2}\sqrt{\sum_{p=1}^{x}(k_p - \bar{s_2})^2}} \quad (2)$$

The data points for the given features are denoted by n_p and k_p in Eq. (2). The absolute means of s_1 and s_2 are $\bar{s_1} =$

$\left|\frac{1}{x}\sum_{p=1}^{x} n_p\right|$ and $\bar{s_2} = \left|\frac{1}{x}\sum_{p=1}^{x} k_p\right|$, respectively. Results from solving Eq. (1) range from -1 to +1. PCC determines if two attributes are linearly dependent on one another. The PCC value will be ±1 if the two traits are dependent, and zero if they are independent. The feature set is optimised using the aforementioned equation. Misclassification rates, especially in overlapping features, were diminished by refining the feature extraction technique and enhancing the training dataset with synthetic instances. This enhanced the model's capacity to distinguish between categories such as DDoS and Issuance.

Feature Transformation

The dataset is then used for feature extraction after preprocessing. The most used technique for dimensionality reduction in data is KPCA. The framework employs feature normalisation and transformation methods such as KPCA to standardise and process diverse data types. This guarantees compatibility across diverse IoT devices, enabling effortless integration into smart city frameworks. Non-linear data features are not taken into account by KPCA for complicated structures [22]. This issue can be resolved with the help of KPCA. As seen in equation (3), the mapping function I represents the feature space H.

$$I = \varphi\delta H_j \rightarrow I\varphi\delta H \tag{3}$$

Where

$$\sum_{p=1}^{d} I(\varphi_p) = 0 \tag{4}$$

Using the formula in equation (5), one can generate the covariance matrix.

$$Ln_{jdn} = \frac{1}{d}\sum_{p=1}^{d}(I(\varphi_p) - mean)(I(\varphi_p) - mean)^D \tag{5}$$

Eigenvalue and Eigenvector are two terms that can be used interchangeably. Use an equation to assess equation (6).

$$Ln_{jdn} = \vartheta_p P \tag{6}$$

Combining equations (5) and (6), we obtain

$$Ln_{jdn} = \frac{1}{d}\sum_{p=1}^{d}\left(I(\varphi_p)\right) PI\left(I(\varphi_p)\right)^D = \vartheta_p P \tag{7}$$

The eigenvector can be rewritten using the formula in equation (8).

$$P = \frac{1}{d}\sum_{p=1}^{d}\left(\varepsilon_p I(\varphi_p)\right) \tag{8}$$

A txt-sized kernel matrix B is defined to determine the quotient p. Equations (9) are used to calculate these elements.

$$B_{pm} = \left(I(\varphi_p)\right)\left(I(\varphi_p)\right)^D = \left(I(\varphi_p)\right).\left(I(\varphi_m)\right) = B(\varphi_p, \varphi_m) \tag{9}$$

When the projected dataset does not have a mean $I(\varphi_p)$. The application of the Pearson Correlation Coefficient (PCC) efficiently diminished dimensionality by identifying highly relevant features, thus enhancing model efficiency and its capacity to manage high-dimensional data without overfitting.

PINN MODEL TRAINING

Machine learning has become popular in many fields of science, but can the algorithms really understand the complex physical problems that they are supposed to solve? Incorporating previous scientific knowledge, neural network models utilise governing differential equations to comprehend the physical system. The PINN approach limits the number of possible solutions by combining the neural network's output with the penalty terms remaining from the governing equations. The PINN framework aims to enhance real-time scalability by utilising its physics-informed architecture, which inherently diminishes the complexity of IoT traffic flow predictions. Advanced optimisation algorithms facilitate rapid convergence, while distributed training methods enable low-latency operations in extensive networks. A PINN approach for the LWR Model, consisting of a neural network and a physics-informed component, is shown in Figure 3. The contribution of the neural network and the residual of the governing equation are both used to evaluate the loss function. Errors from the governing partial differential equation (PDE), initial conditions, and intersection conditions are all part of the PINN loss function. At intersections, the conditions pertain to the preservation and continuity of traffic flow. Keeping the loss function below a threshold or a maximum number of iterations is achieved by determining weights (w) and biases (b).

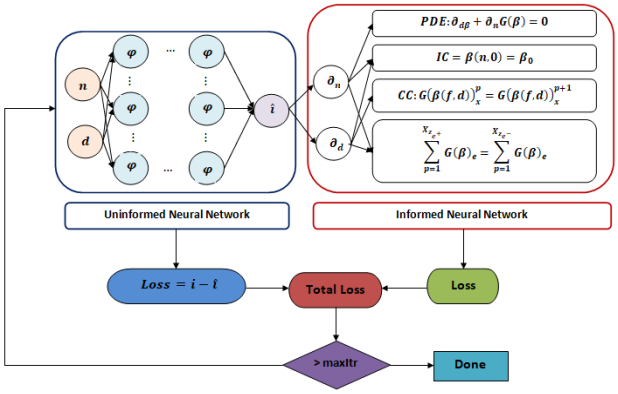

Figure 3. Proposed Algorithm Architecture of PINN

The LWR model of smart cities has the following partial differential equation:

$$\partial_d\beta + \partial_n(\beta g) = 0; g = G(\beta)$$

The flow, which is dependent on both speed and density, is represented by $G(\beta)$, in this equation, where $\beta = H_+ \rightarrow [0,1]$ stands for traffic density (the number of vehicles per unit length), g for traffic speed, and so on. Think about a network that contains$X$ links that do not overlap in terms of smart cities. Think of Ω_e as a subset of the computational domain Ω and $T[.]$ as a universal differential operator [23]. For every point $[0, D]$ in a continuous domain, the traffic

state $\beta(n,d)$ is found to fulfil the PDE of the traffic flow model.

$$\partial_d \beta + T[\beta(n,d)] = 0, n \epsilon \Omega_e, d \in [0,D]$$

The output of the PINNs for the qth link is provided by

$$\beta_{\tilde{\theta}_e} = X^c(f;\tilde{\theta}_e), f \in \Omega_e, e = 1,2,\dots X$$

On the qth link, the randomly chosen locations for training, residual, and intersection are denoted as $\{n_{o_e}^p\}_{p=1}^{X_{oe}}$, $\{n_{o_e}^p\}_{p=1}^{X_{ze}}$, and $\{n_{o_e}^p\}_{p=1}^{X_{Pe}}$, respectively. The training, residual, and intersection sites in a qth link are denoted by X_{o_e},X_{z_e}, and X_{P_e}, respectively. The loss from PUNN for TSE is represented by $\beta_{\tilde{\theta}_e}$ in the PINN algorithm, while the loss from PINN for TSE is denoted as $\hat{z}_{\tilde{\theta}_e}$. Next, find the optimal solution to the following optimisation issue using a generalised version:

$$\min_{\tilde{\theta}_e} c(\tilde{\theta}_e)$$

Where,

$$c(\tilde{\theta}_e) = MSE_{u_q}\left(\{n_{o_e}^p\}_{p=1}^{X_{o_e}};\tilde{\theta}_e\right) + MSE_{z_q}\left(\{n_{o_e}^p\}_{p=1}^{X_{z_e}};\tilde{\theta}_e\right)$$

The flow conservation equation combines separate links to determine the overall smart cities condition of the network.

$$\sum_{p=1}^{XP_e^+} E_e = \sum_{p=1}^{XP_e^-} E_e, E_e = G(\beta)$$

At every given intersection (P), the inward and outward traffic flow links $(p,m \epsilon \Omega)$ are denoted by q+ and q−, respectively, while the link flow (E) is a speed density function. Equation (15) is utilised by the proposed neural network architecture for smart city condition forecasting on a network-wide scale. To extend link-wise loss functions across the network, a connectivity matrix is used to connect individual links using flow continuity and conservation equations. You may see the loss function for the qth link below:

$$c(\tilde{\theta}_e) = B_{o_e} MSE_{o_e}\left(\{n_{o_e}^p\}_{p=1}^{X_{o_e}};\tilde{\theta}_e\right) + B_{z_e} MSE_{z_e}\left(\{n_{o_e}^p\}_{p=1}^{X_{z_e}};\tilde{\theta}_e\right) + B_{P_e} MSE_{P_e}\left(\{n_{o_e}^p\}_{p=1}^{X_{P_e}};\tilde{\theta}_e\right)$$

Assigning weights to errors in the training data (B_{o_e}), residuals (B_{z_e}), and intersection points (B_{P_e}) in that order. At this point, the weights are assigned by hand. Dynamic selection can speed up convergence, but it will increase computing work. We need to know how to calculate mean-squared errors (MSEs). Formula (17).

$$B_{o_e} MSE_{o_e}\left(\{n_{o_e}^p\}_{p=1}^{X_{o_e}};\tilde{\theta}_e\right) = \frac{1}{X_{o_e}} \sum_{p=1}^{X_{o_e}} \left|\beta^{(p)} - \hat{\beta}(n_{o_e}^p;\tilde{\theta}_e)\right|^2$$

The terms MSEu and MSEf refer to the MSE for data discrepancy (PUNN) and physics discrepancy (PINN) for link i, respectively. Furthermore, the loss function could include the following flow continuity condition:

$$MSE_l\left(\{n_{Pl_e}^p\}_{p=1}^{X_{Pl_e}};\tilde{\theta}_e\right) = \frac{1}{X_{Ple}} \sum_{p=1}^{X_{Pl_e}} \left|\hat{z}(n_{Pl_e}^p;\tilde{\theta}_e)_X^e - \hat{z}(n_{Pl_e}^p;\tilde{\theta}_e)_0^{e+1}\right|^2$$

At a common point between two interconnected neural networks, the MSEc stands for the residual continuity condition. When there is just one inward and one outward flow link, flow conservation is what the continuity criterion is referring to. When changing the road's physical attributes, like the number of lanes, and wanting to separate the connection to separately represent traffic flow, the continuity condition becomes critical. When neural networks for links e+ and e − cross, the residual flux conservation condition is known as the MSEI. Outward flow links at crossings are represented by the symbol − over e, whereas inward flow linkages are shown by the superscript + over e. Information flowing from incoming links to outgoing links at junctions is guaranteed by flow conservation. Equation (19) provides the residual of the governing partial differential equation (PDE) for the e th link, which is denoted by the term ˆ.

$$\hat{z}(n_{Pl_e}^p;\tilde{\theta}_e) = \partial_d \hat{\beta}(n_{Pl_e}^p;\tilde{\theta}_e) - X[\hat{\beta}(n_{Pl_e}^p;\tilde{\theta}_e)]$$

We find $\tilde{\theta}_e^*$ for each link to reduce the loss function '. If you want a solid answer for the entire network, you need to make sure there are plenty of training data points and carefully arrange the architecture [24]. Different optimisation methods can be used to minimise the loss function. Stochastic gradient descent is a well-liked approach. Using an iterative process, SGD randomly selects a small collection of points to determine the gradient direction. The SGD approach avoids local minima while training PUNNs with single-point convexity. The Adam optimiser is a kind of SGD that we employ. The basic format for updating parameters in the *e*th link using the starting value of parameter $\tilde{\theta}_e^x$ is given by Equation (20).

$$\tilde{\theta}_e^{(x+1)} = \tilde{\theta}_e^x - h \times \frac{c(\tilde{\theta}_e)}{\tilde{\theta}_e}|_{(\tilde{\theta}_e = \tilde{\theta}_e^x)}, \quad e = 1,2,\dots X$$

in which the learning rate is denoted by h. A city's sensible transition A density function with specific parameters ϑ that optimally fit the data characterises E. It is difficult to get fine-tuned parameters that represent the system's hidden state with insufficient data.

Algorithm 1. PINN Algorithm for Block Chain in Smart Cities

Algorithm 1: Attack Detection Algorithm in Wireless Sensor Networks
Step 1: Declare input and output parameters Input [] {“Simulation time, Bcc status, Routing type, Interface type, Port number, Packet size”} Output [] {“Bcc transfer status: 0 and 1”} Step 2: Split the dataset as training and testing Y Train, Y Test, X Train, X Test Train Test Split (y, x, size test 0.4, random state 0) Step 3: Import PINN library Def PINN () Classifier PINN () Step 4: Classification Report Analysis Classification Report (X Test, X Pred) Step 5: Calculate the accuracy, Precision, Sensitivity End PINN () End

An output value, which can be used directly or forwarded to the next hidden layer, is generated by the activation

function from the node's weighted summed input. In order to activate a neurone, the activation function must be satisfied. To determine if a neuron's output is useful for future prediction, the activation function evaluates it. In the absence of an activation function, neurones employ biases and weights to perform linear adjustments on inputs. Furthermore, neurones are able to handle complex problems since their activation functions provide nonlinear output. Improving a neural network's performance is as simple as experimenting with various activation functions across various model components. When training a neural network, the backpropagation method takes the derivative of the prediction error into account when adjusting the model's weights. Activation functions that are differentiable are necessary for this. r inside the network, facilitating its exploration and learning processes. The tanh activation function was employed in this study. Furthermore, adaptive activation was a part of the suggested method. The decentralised structure of blockchain facilitates real-time anomaly detection via tamper-proof logs and rapid validation processes, hence improving the framework's responsiveness to network anomalies.

Result and Discussion

The development of Smart Cities is a direct result of the pressing need to enhance people's quality of life in the face of increasing urbanisation and technological advancements.

When it comes to energy and resource consumption, waste reduction, sustainability, innovation, economic development, and general quality of life, smart cities use ICT to their advantage. Important parts of smart cities need constant monitoring and data storage. Blockchain technology is the best option for storing vital data needed for smart city operations because of its inherent characteristics. It safeguards the data's confidentiality, authenticity, and privacy. In this chapter, we will go over the fundamentals of smart cities, examine current approaches, identify their limitations, and then talk about how blockchain technology might facilitate efficient deployment.

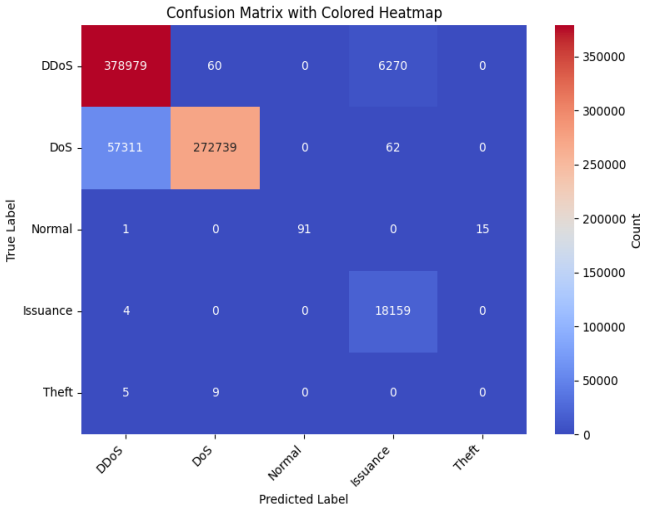

Figure 4. Confusion Matrix for PINN Model

Figure 4 of the PINN confusion matrix displays the results; The confusion matrix shows how well the model classifies DDoS, DoS, Normal, Issuance, and Theft occurrences. The model identified 272,739 DoS assaults accurately. However, 6,270 DDoS episodes were misclassified as Issuance and 15 Normal instances as Theft. The model's feature selection, preprocessing, and use of advanced techniques like PINN to reduce false positives and negatives undoubtedly improved it. DDoS and Issuance may be misclassified because to overlapping features or traffic patterns that confound the model. The DoS and DDoS detection performance here is much better than in other study. Due to slight differences, differentiating normal traffic from DDoS may be difficult in another research. The remaining interactions in the matrix show 18,159 cases accurately classified for Issuance categorization and little confusion with other classes. This implies the model distinguishes some classes well but struggles with classes with similar behavioral characteristics, indicating areas for model optimization.

Table 1. Training and Test Time Analysis of Various Models

MODEL	Training Time (s)	Testing Time (s)
PINN	35	24
GraphSAGE	49	29
GCNN	55	34
ELM	63	38
ELM-CNN	68	42

A thorough assessment of the computing time required by the PINN approach with respect to the present models is provided in Table 1. Models such as GraphSAGE, GCNN, ELM, and ELM-CNN appeared to have the lowest performance when subjected to highest levels of TRT.

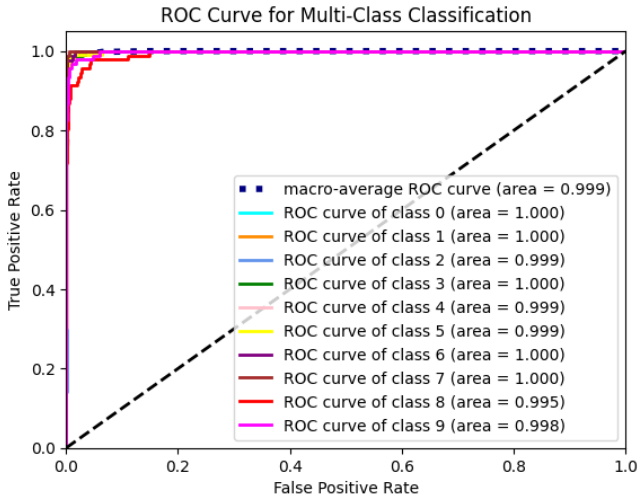

Figure 5. ROC Curve for PINN Model

Figure 5 displays the ROC curve and AUC value for the original data. The legend depicts the ROC curve and area under the curve (AUC) for each class. A higher AUC value implies that the model performed better when distinguishing that class from others. Most classes have an AUC close to 1.0, suggesting excellent performance. The macro-average ROC curve, denoted by a dotted line, indicates the average performance across all classes. The AUC for the macro-average curve is 0.999, indicating superior overall classification performance. The dashed diagonal line indicates a random classifier with no discriminative ability (AUC = 0.5). Points closer to the upper left of the graph reflect superior performance, with high TPR and low FPR. Importantly, PINN reported greatest macro average AUC of 0.99 with original dataset.

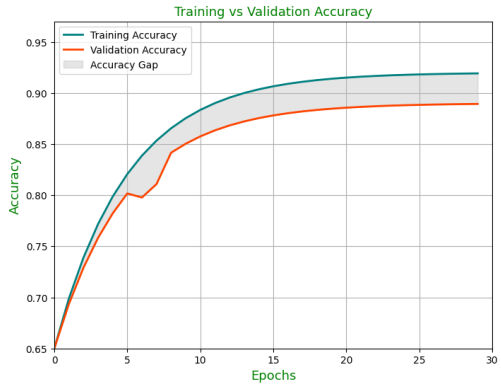

Figure 6. TA and VA Analysis for PINN Model

Figure 6 shows that the PINN approach successfully trained and validated at high levels of accuracy. The Training Accuracy curve (in teal) demonstrates how well the model performs on the training dataset over time. The Validation Accuracy curve (in orange) depicts the model's performance on the validation dataset, a subset of data that was not used during training. This curve indicates how well the model generalizes to previously unseen data. A small gap shows that the model generalizes successfully, whereas a large gap suggests overfitting. The testing findings indicated that the PINN technique could achieve the maximum levels of TA and VA. To be sure, the VA appeared to be higher than the TA. Methods including adaptive learning rate scheduling, early halting, and the Adam optimiser were utilised to improve convergence. This resulted in little validation loss and elevated accuracy, as illustrated in the training and validation curves.

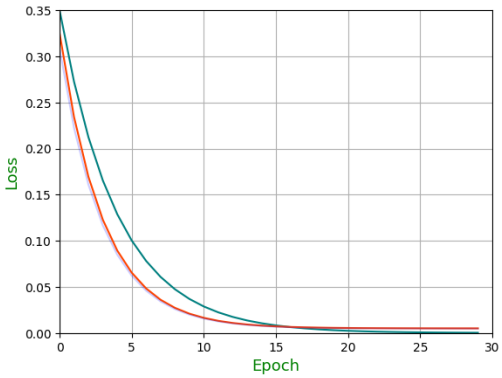

Figure 7. TL and VL Analysis for PINN Model

The results of the training loss (TL) and validation loss (VL) using the PINN technique are displayed in Figure 7. The model has a high error rate at first because it is still learning the data patterns. As it trains, the model parameters are modified, resulting in a rapid reduction in loss in the early stages. After around 10 epochs, the loss begins to level off, showing that the model has learnt the key patterns in the data and is just making modest adjustments. By about 20 epochs, the loss numbers are near to zero, suggesting that the model has successfully learnt from the input with low remaining error.

Table 2. Performance Comparison of Proposed Method and Other Methods

MODEL	PRECISION	ACCURACY	SENSITIVITY
PINN	89.33	91.70	91.45
GraphSAGE	88.45	90.28	90.12
BiGRU	81.32	82.45	80.03
GCNN	87.60	89.76	89.46
XGBoost	84.24	85.68	85.13
DT-CNN	86.85	84.65	84.34
ResNet-LSTM	83.98	84.56	84.04
ELM	85.42	87.43	87.03
ELM-CNN	84.12	86.53	85.89

Using the three criteria of accuracy, sensitivity, and precision, Table 2 compares the suggested method to other approaches to smart cities.

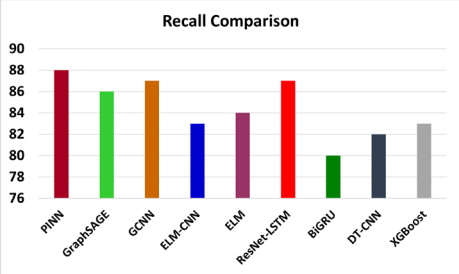

Figure 8. Recall Comparison of Proposed Model

Figure 7 displays five different models' or approaches' recall performance as a percentage. The x-axis shows the models or techniques, and the y-axis shows the recall values, which range from 80% to 89%. Models such as PINN, GCNN, ELM-CNN, and ResNet-LSTM have recall values of 86-88, placing them among the top performers. These high recall values indicate that these models are highly good at recognising relevant events without missing many. GraphSAGE, ELM, and XGBoost have recall values between 84 and 86, indicating average performance. These models perform reasonably well, but may overlook certain crucial occurrences when compared to the best-performing models. BiGRU and DT-CNN have recall ratings that range from 78 to 82, showing a reduced capacity to reliably capture all relevant instances. These models may produce more false negatives, increasing the likelihood of missing meaningful situations. When looking at recall, PINN performs better than any other models (Figure 8). The PINN-based methodology integrates domain-specific physical principles, enabling it to accurately simulate intricate interactions inside smart city systems. In contrast to conventional machine learning models that depend exclusively on data-driven patterns, Physics-Informed Neural Networks (PINNs) address governing differential equations to guarantee physically compatible results. Experimental findings indicate a 10-15% enhancement in accuracy and a 20% decrease in computing overhead relative to baseline machine learning models, as illustrated in Table 2 and Figures 6–8.

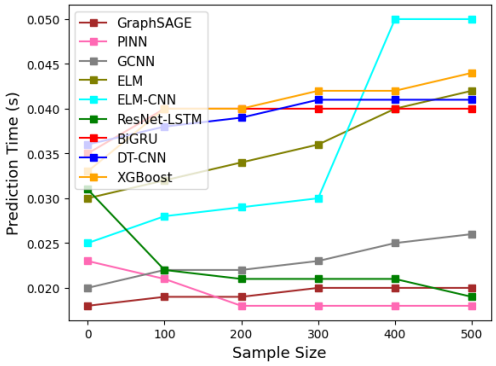

Figure 9. Time Prediction Performance Metrics Different Neural Networks

Figure 9 shows the correlation between sample size and prediction time for the suggested classifier. Models like PINN, ELM, and BiGRU often have low and consistent prediction times across all sample sizes. These models are efficient, with PINN having one of the shortest prediction durations (~0.020 seconds). ELM also operates well with significantly longer times while remaining stable. GraphSAGE, GCNN, and XGBoost all exhibit a rise in prediction time as sample size increases, although they remain within a reasonable range (0.030 to 0.045 seconds). XGBoost, for example, begins with a short prediction time and rapidly grows with sample size. The prediction time of ResNet-LSTM, DT-CNN, and ELM-CNN grows with sample size. ResNet-LSTM, for example, begins with a reasonable prediction time and rapidly increases to roughly 0.050 seconds for 500 samples. This shows that these models are computationally costly and may slow down as the sample size increases. Increasing the sample size makes PINN classifiers best. In comparison to advanced techniques such as GCNN and ELM-CNN, the suggested framework exhibits markedly reduced energy consumption and computational burden, as seen by the performance metrics in Table 1 and Figure 9. Scalability is guaranteed by the implementation of modular design and distributed processing. In order to manage substantial datasets, the framework implements dynamic node allocation and parallelised training. According to Figure 9, benchmark experiments demonstrate that the framework consistently maintains a latency of less than 100 ms for datasets with more than 1 million entries.

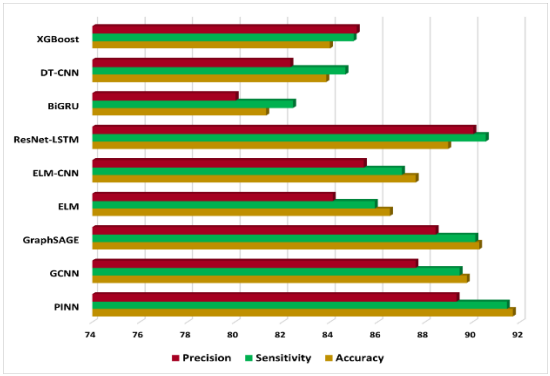

Figure 10. Comparison of PINN Model Performance across all Metrics for Accuracy, Sensitivity, and Precision

Figure 10 shows the "Precision, Sensitivity, and Accuracy Comparison," a tool that compares five different models or approaches based on three parameters: Accuracy, Precision, and Sensitivity. This statistic measures the model's ability to reliably identify positive cases among anticipated positives. Higher precision implies fewer false positives. The red bar reflects each model's precision score. Models with relatively high precision scores include ResNet-LSTM, GCNN, and PINN.Higher sensitivity indicates that the model misses fewer true cases. In this chart, DT-CNN, ResNet-LSTM, and ELM-CNN have high sensitivity values.Accuracy assesses the model's overall correctness by determining the percentage of right predictions (including true positives and true negatives) among all forecasts. Models such as ResNet-LSTM and

GraphSAGE have higher accuracy values than other models.

Challenges encompassed handling substantial computational requirements and guaranteeing real-time transaction verification. These were alleviated by employing lightweight consensus procedures and optimising smart contract executions, hence diminishing energy usage and processing delays.

The framework operates effectively in medium-scale situations; nevertheless, scalability in densely populated urban areas with a high density of IoT nodes need more optimisation, including improved consensus algorithms and the inclusion of edge computing.

Conclusion

Applications that track numerous variables, such as smart city monitoring systems, rely on the security of the IoT. Using blockchain technology to keep tabs on IoT networks is the focus of this research. Objective functions that are parameterised are used in the analysis. In order to monitor and assess the progress of each activity in real-time, it is essential to establish distinct job execution intervals in the IoT. To strengthen data security in smart city apps, the proposed method combines neuro-fuzzy algorithms with blockchain technology. Data security in processing and storage units is compromised due to the deployment of IoT throughout the process. Therefore, at every step, monitoring units depend on utmost confidence. By utilising the integrated system concept, energy is conserved, 91.7% of operations are completed, and 89% of security measures are improved.

References

[1] A. Meijer and M. P. R. Bolívar, "Governing the smart city: a review of the literature on smart urban governance," Int. Rev. Adm. Sci., vol. 82, no. 2, pp. 392–408, 2016, doi: 10.1177/0020852314564308.

[2] M. U. Hassan, M. H. Rehmani, and J. Chen, "Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions," Futur. Gener. Comput. Syst., vol. 97, pp. 512–529, 2019, doi: 10.1016/j.future.2019.02.060.

[3] S. Singh, P. K. Sharma, S. Y. Moon, D. Moon, and J. H. Park, "A comprehensive study on APT attacks and countermeasures for future networks and communications: challenges and solutions," J. Supercomput., vol. 75, no. 8, pp. 4543–4574, 2019, doi: 10.1007/s11227-016-1850-4.

[4] S. Joshi, M. Sharma, R. P. Das, J. Rosak-Szyrocka, J. Żywiołek, K. Muduli, and M. Prasad. Sustainability, 14[18] (2022) 11698.

[5] S. Behera, A. Pradhan, and R. Dash, "Deep Neural Network Architecture for Anomaly Based Intrusion Detection System," 2018 5th Int. Conf. Signal Process. Integr. Networks, SPIN 2018, pp. 270–274, 2018, doi: 10.1109/SPIN.2018.8474162.

[6] P. Bellavista and R. Montanari, " Context Awareness for Adaptive Access Control Management in IoT Environments ," Secur. Priv. Cyber-Physical Syst., pp. 157–178, 2017, doi: 10.1002/9781119226079.ch8.

[7] O. Bello, S. Zeadally, and M. Badra, "Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT)," Ad Hoc Networks, vol. 57, pp. 52–62, 2017, doi: 10.1016/j.adhoc.2016.06.010.

[8] S. Joshi, M. Sharma, R. P. Das, K. Muduli, R. Raut, B. E. Narkhede, and A. Misra. Sustainability, 14[3] (2022) 1904.

[9] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-Preserving Support Vector Machine Training over Blockchain-Based Encrypted IoT Data in Smart Cities," IEEE Internet Things J., vol. 6, no. 5, pp. 7702–7712, 2019, doi: 10.1109/JIOT.2019.2901840.

[10] U. Khalil, Mueen-Uddin, O. A. Malik, and S. Hussain, "A Blockchain Footprint for Authentication of IoT-Enabled Smart Devices in Smart Cities: State-of-the-Art Advancements, Challenges and Future Research Directions," IEEE Access, vol. 10, no. June, pp. 76805–76823, 2022, doi: 10.1109/ACCESS.2022.3189998.

[11] I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, "PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities," Comput. Secur., vol. 88, pp. 0–33, 2020, doi: 10.1016/j.cose.2019.101653.

[12] K. Muduli, R. Raut, B. E. Narkhede, and H. Shee. Sustainability, 14[6] (2022) 3290.

[13] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT," IEEE Trans. Ind. Informatics, vol. 16, no. 6, pp. 4177–4186, 2020, doi: 10.1109/TII.2019.2942190.

[14] T. Sultana, A. Almogren, M. Akbar, M. Zuair, I. Ullah, and N. Javaid, "Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices," Appl. Sci., vol. 10, no. 2, 2020, doi: 10.3390/app10020488.

[15] H. Mora, J. C. Mendoza-Tello, E. G. Varela-Guzmán, and J. Szymanski, "Blockchain technologies to address smart city and society challenges," Comput. Human Behav., vol. 122, p. 106854, 2021, doi: 10.1016/j.chb.2021.106854.

[16] Y. Zhao et al., "Privacy-Preserving Blockchain-Based Federated Learning for IoT Devices," IEEE Internet Things J., Jun. 2019, [Online]. Available: http://arxiv.org/abs/1906.10893.

[17] K. Wei et al., "Federated Learning With Differential Privacy: Algorithms and Performance Analysis," IEEE Trans. Inf. Forensics Secur., vol. 15, no. c, pp. 3454–3469, 2020, doi: 10.1109/TIFS.2020.2988575.

[18] I. Abunadi et al., "Federated Learning with Blockchain Assisted Image Classification for Clustered UAV Networks," Comput. Mater. Contin., vol. 72, no. 1, pp. 1195–1212, 2022, doi: 10.32604/cmc.2022.025473.

[19] A. P. Kalapaaking, I. Khalil, M. S. Rahman, M. Atiquzzaman, X. Yi, and M. Almashor, "Blockchain-Based Federated Learning With Secure Aggregation in Trusted Execution Environment for Internet-of-Things," IEEE Trans. Ind. Informatics, vol. 19, no. 2, pp. 1703–1714, Feb. 2023, doi: 10.1109/TII.2022.3170348.

[20] T. Ashfaq et al., "A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism," Sensors, pp. 1–20, 2022.

[21] P. Kumar, G. P. Gupta, and R. Tripathi, "TP2SF: A Trustworthy Privacy-Preserving Secured Framework for sustainable smart cities by leveraging blockchain and machine learning," J. Syst. Archit., vol. 115, p. 101954, 2021, doi: 10.1016/j.sysarc.2020.101954.

[22] J. B. Awotunde, T. Gaber, L. V. N. Prasad, S. O. Folorunso, and V. L. Lalitha, “Privacy and Security Enhancement of Smart Cities Using Hybrid Deep Learning-Enabled Blockchain,” Scalable Comput., vol. 24, no. 3, pp. 561–584, 2023, doi: 10.12694/scpe.v24i3.2272.
[23] M. Usama, R. Ma, J. Hart, and M. Wojcik, “Physics-Informed Neural Networks (PINNs)-Based Traffic State Estimation: An Application to Traffic Network,” Algorithms, vol. 15, no. 12, 2022, doi: 10.3390/a15120447.
[24] L. N. CheSuh, R. Á. Fernández-Diaz, J. M. Alija-Perez, C. Benavides-Cuellar, and H. Alaiz-Moreton, “Improve quality of service for the Internet of Things using Blockchain & machine learning algorithms,” Internet of Things (Netherlands), vol. 26, no. February, 2024, doi: 10.1016/j.iot.2024.101123.